In the claims:

1. (previously presented) A method for determining a steady-state operating point of a first network node relative to a second network node, the method comprising:

determining a queue function of the first node based upon predetermined system traffic conditions; and

determining a control function for the second node based upon the queue function,

wherein the control function prompts a gradual increase of drop probability in an overload condition.

2. (original) A method according to claim 1, wherein the control function is a random early detection control function.

3. (original) A method according to claim 1, wherein the control function does not have a non-bounded discontinuity.

4. (original) A method according to claim 1, wherein the control function comprises two piecewise linear segments.

5. (previously presented) A method for modeling dynamics of a queue in a first node having a buffer in order to determine a steady-state operating point of the first node relative to a second node, the method comprising:

calculating a queue law function dependent on traffic conditions at the second node; and

determining a point of operation for the first node as the intersection of the queue law function and a predetermined control function for the first node,

wherein the control function increases drop probability gradually in an overload condition.

6. (original) A method according to claim 5, wherein the point of operation defines a packet drop percentage for dropping a percentage of packets from the buffer.

7. (original) A method for improving congestion control according to claim 5, wherein the node resides in a network.

8. (original) A method for improving congestion control according to claim 7, wherein the network operates in a TCP environment.

9. (original) A method for improving congestion control according to claim 5, wherein data received at a node is acknowledged.

10. (original) A method according to claim 5, wherein the traffic conditions which determine the queue law function are the number of flows into the node, an average packet size and an average round trip transmission time.

11. (original) A method according to claim 5, wherein the function is determined by assuming that the node does not experience feedback.

12. (original) A method according to claim 5, wherein the point of operation determines a drop rate.

13. (original) A method according to claim 5, further comprising:

dropping packets from the buffer at the determined drop rate.

14. (previously presented) A method for defining an average queue size function for a first node having a buffer of a given size, in a network in which data sent from the first node through a link which, when received by a second node, is acknowledged by the second node, in order to determine a steady-state operating point of the first node relative to the second node, the method comprising:

determining a quantity that is representative of the link utilization between the first and second nodes;

calculating a quantity that is representative of an average round trip transmission time for data to be sent from the first node to the second node and an acknowledgment to be received by the first node; and

calculating the average queue size function dependent on a data drop probability based upon the link utilization, the buffer size, and the average round trip transmission time, the drop probability increasing gradually in an overload condition.

15. (original) A method according to claim 14, wherein the average queue size function is dependent upon the number of flows through the queue and an average packet size.

16. (original) A method for defining an average queue size function according to claim 15, further comprising:

predicting a drop probability based in part upon the average queue size function.

17. (original) A method according to claim 16, wherein the average queue size function is dependent upon the number of flows through the queue and an average packet size.

18. (previously presented) A method for estimating an average queue size for a first node having a buffer with a queue wherein the first node resides on a link, in order to determine a steady-state operating point of the first node relative to a second node, the method comprising:

determining a round trip transmission time for the link; and

determining the average queue size at the intersection point of a node congestion control function and a queue law function, wherein the queue law function is based in part on the round trip transmission time, and wherein the control function defines a drop probability which increases gradually in an overload condition.

19. (previously presented) A method for designing a control function for use in a congestion control module residing in a network, in order to determine a steady-state operating point of a first node relative to a second node, the method comprising:

determining a maximum average queue size function based at least upon a minimum value for the average round trip transmission time;

selecting a point defining a maximum value for the control function outside of the maximum average queue size function; and

defining the control function as being bounded by the maximum value and crossing the maximum average queue size function, the drop probability increasing gradually in an overload condition.

20. (original) A method according to claim 19, wherein the step of defining a function includes selecting a linear equation as the control function wherein the linear function passes through the maximum value point.

21. (original) A method according to claim 19, wherein the selection of the point is also dependent on a queue management policy.

22. (original) A method according to claim 19, wherein the maximum control function is dependent upon line speed for the network.

23. (original) A method according to claim 19, wherein the congestion control module is based on random early detection.

24. (original) A method for defining an average queue size function according to claim 19, further comprising:

predicting a drop probability based in part upon the average queue size function in a congestion control module of the first link.

25. (previously presented) A method for determining parameters used a random early detection congestion control module residing in a first node in a network, in order to determine a steady-state operating point of the first node relative to a second node, the method comprising:

receiving input parameters including a line speed for the node;

calculating values including a buffer size for an input to the link, a queue sampling interval, and an average weight; and

employing the values and parameters to define a drop probability which increases gradually in an overload condition.

26. (original) A method according to claim 25, wherein the values are used to determine values for qmin, pmin, qmax and pmax.

27. (previously presented) A method for determining the minimum buffer size in a congestion control module having a control function in a TCP network defined by a queue law, in order to determine a steady-state operating point of a first node relative to a second node, comprising the steps of:

determining an equilibrium point where the control function and the queue law intersect; and

selecting a buffer size that is larger than the average queue size at the intersection point, the drop probability being defined by the control function and increasing gradually in an overload condition beginning at the intersection.

28. (previously presented) A method for creating a stable queue control function for managing a queue in a node within a network, wherein the queue control function determines a packet drop rate based upon an average queue size, the method comprising:

calculating a maximum queue law function based on traffic conditions for the network and designating a maximum boundary for expected operating conditions of the queue control function to be outside of the maximum queue law function, the control function defining a drop probability which increases gradually in an overload condition.

29. (original) A method according to claim 28, wherein the queue control function is a random early detection control function.

30. (previously presented) A method for setting a value for the maximum boundary point for expected operating conditions for a congestion control function in a network, the method comprising:

selecting a queue management policy;

determining a maximum average queue size for expected operating conditions based upon the selected queue management policy;

selecting a corresponding value for the drop rate to be any point that lies outside of a queue law function for the network,

the control function defining a drop probability which increases gradually in an overload condition.

31. (original) A method according to claim 30, wherein the queue management policy is a drop conservative policy.

32. (original) A method according to claim 30, wherein the queue management policy is a delay conservative policy.

33. (original) A method according to claim 31, wherein the maximum average queue size for normal operating conditions is determined by evaluating the maximum queue law function.

34. (previously presented) A method according to claim 33, Wherein qmax is determined by multiplying the maximum delay by the line speed.

35. (previously presented) A method according to claim 30, wherein the maximum average queue size is determined by evaluating an inverse of the maximum queue law function using the maximum average queue size.

36. (previously presented) A method of determining a minimum buffer size in a congestion control module wherein the congestion control module drops packets within a buffer based upon a congestion control function, the method comprising:

selecting a value for a maximum drop probability; and

evaluating a maximum queue law function using the maximum drop probability to determine qmax, the minimum buffer size; and

employing the values and parameters to define a drop probability which increases gradually in an overload condition.

37. (previously presented) A method for determining a weighing factor for a queue estimator, wherein the queue estimator calculates the average queue size based on a moving average of samples, the method comprising:

determining a sampling period;

determining a sample value defining when a sample's contribution to the average queue size is negligible;

determining a total time value for total time for all samples that contribute to the average queue size; and

evaluating the weight based upon the sample value, the sampling period and the total time value, the calculated average queue size being employed to define a drop probability which increases gradually in an overload condition.

38. (previously presented) A method according to claim 37, wherein the step of evaluating the weight includes utilizing an equation: weight =1-sample value$^{\text{sampling period/total time value}}$.

39. (previously presented) A method for designing a stable congestion control function for use in a congestion control module in a network, the method comprising:

determining a maximum queue law function based upon maximum expected traffic conditions;

when the maximum queue law function is placed on a graph having drop rate percentage and average queue size for axes, selecting a point outside of the maximum queue law; and

selecting a function to be the control function that is bounded at the selected

point, the control function defining a drop probability which increases gradually in an overload condition.

40. (previously presented) A method for estimating average queue size of a queue in a buffer within a congestion control module in a network, the method comprising:

periodically sampling the queue size of the buffer;

using a queue estimator in conjunction with each periodically sampled queue size to determine an average queue size; and

periodically updating the average queue size based upon a point of intersection of a maximum queue law function and a control function of the congestion control module, the control function defining a drop probability which increases gradually in an overload condition.

41. (original) A method according to claim 40, wherein the maximum queue law function is determined based upon current traffic conditions within the network.

42. (previously presented) A systematic method for determining a weighting factor for use in calculating average queue size of a buffer in a node in a network wherein packeted data is sent from one node to another node at a sending rate and wherein a protocol used in the network increases the packet sending rate so long as each packet is acknowledged, the method comprising:

selecting a sampling interval wherein the sampling interval is at most equal to a packet roundtrip time;

determining a total time interval for which samples contribute to the average queue size based on a time period for which the sending rate increases for the network;

calculating the weight based upon the sampling interval and the total time interval, the calculated average queue size being employed to define a drop probability which increases gradually in an overload condition.

43. (previously presented) An apparatus for determining a control function wherein the control function is used in a congestion control module in a network, the apparatus comprising:

a queue module for determining a queue function based upon predetermined system parameters; and

a control function module for determining the control function based upon the queue function, the control function defining a drop probability which increases gradually in an overload condition.

44. (previously presented) An apparatus for modeling dynamics of a queue in a node having a buffer, the method comprising:

a queue module for calculating a queue function dependent on traffic conditions at the node; and

a processor for determining a point of operation for the node as the intersection of the queue law function and a predetermined control function for the node, the control function defining a drop probability which increases gradually in an overload condition.

45. (original) An apparatus according to claim 43, wherein the control function is a random early detection control function.

46. (original) An apparatus according to claim 43, wherein the control function does not have an undefined point.

47. (original) An apparatus according to claim 43, wherein the control function comprises two piecewise linear segments.

48. (original) An apparatus according to claim 44, wherein the point of operation defines a packet drop percentage for dropping a percentage of packets from the buffer.

49. (original) An apparatus according to claim 44, wherein the node resides in a network.

50. (original) An apparatus according to claim 49, wherein the network operates in a TCP environment.

51. (original) An apparatus according to claim 44, wherein the traffic conditions which determine the queue law function are number of flows into the node, an average packet size and an average round trip transmission time.

52. (original) An apparatus according to claim 44, wherein the point of operation determined by the processor determines a drop rate.

53. (original) An apparatus according to claim 52, wherein the processor drops packets from the buffer at the determined drop rate.

54. (previously presented) An apparatus for determining control function configuration parameters for designing a control function for use in a congestion control module residing in a network, the apparatus comprising:

a configuration module receiving as input system parameters and outputting control function configuration parameters based upon a maximum average queue size function, the control function defining a drop probability which increases gradually in an overload condition.

55. (original) An apparatus according to claim 54, wherein at least one of the control function configuration parameters is determined as a point residing outside of the maximum average queue size function.

56. (original) An apparatus according to claim 55, wherein the at least one of the control function configuration parameters is dependent on a selected queue management policy.

57. (previously presented) An apparatus for determining a weight for estimating an average queue size in a queue estimator for a node the apparatus comprising:

a weight calculation module for receiving input parameters including a line speed for the node wherein the weight calculation module calculates a queue sampling interval and uses the queue sampling interval to calculate the weight, the estimated average queue size being employed to define a drop probability which increases gradually in an overload condition.

58. (previously presented) An apparatus for determining the minimum buffer size in a congestion control module having a control function in a TCP network defined by a queue law, the apparatus comprising:

a configuration module for determining an equilibrium point where the control function and the queue law intersect, the control function defining a drop probability which increases gradually in an overload condition; and

an input selector allowing for selection of the minimum buffer size so that the minimum buffer size is larger than the average queue size at the intersection point.

59. (previously presented) An apparatus for estimating average queue size of a queue in a buffer within a congestion control module in a network, the apparatus comprising:

a sampler for obtaining periodic samples of the queue size of the buffer;

a queue estimator for use in conjunction with each periodically sampled queue size to determine an average queue size;

a processor for periodically updating the average queue size based upon a point of intersection of a maximum queue law function and a control function of the congestion control module, the control function defining a drop probability which increases gradually in an overload condition.

60. (original) An apparatus according to claim 59, wherein the maximum queue law function is determined based upon current traffic conditions within the network.

61. (previously presented) A computer program product for determining a control function for use with a computer wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for determining a queue function based upon predetermined system parameters; and

computer code for determining the control function based upon the queue function, the control function defining a drop probability which increases gradually in an overload condition.

62. (previously presented) A computer program product for modeling dynamics of a queue in a node having a buffer, wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for calculating a queue function dependent on traffic conditions at the node; and

computer code for determining a point of operation for the node as the intersection of the queue law function and a predetermined control function for the node thereby defining a drop probability which increases gradually in an overload condition.

63. (original) A computer program product according to claim 61, wherein the control function is a random early detection control function.

64. (cancelled) A computer program product according to claim 61, wherein the control function does not an indefinite point.

65. (original) A computer program product according to claim 61, wherein the control function comprises two piecewise linear segments.

66. (original) A computer program product according to claim 62, wherein the point of operation defines a packet drop percentage for dropping a percentage of packets from the buffer.

67. (original) A computer program product according to claim 62, wherein the node resides in a network.

68. (original) A computer program product according to claim 67, wherein the network operates in a TCP environment.

69. (original) A computer program product according to claim 62, wherein data received at a node is acknowledged.

70. (original) A computer program product according to claim 62, wherein the traffic conditions which determine the queue law function are the number of flows into the node, an average packet size and an average round trip transmission time.

71. (original) A computer program product according to claim 62, wherein the function is determined by assuming that the node does not experience feedback.

72. (original) A computer program product according to claim 62, wherein the point of operation determines a drop rate.

73. (original) A computer program product according to claim 62, further comprising: computer code for dropping packets from the buffer at the determined drop rate.

74. (previously presented) A computer program product for defining an average queue size function for a first node having a buffer of a given size in a network in which data sent from the first node through a link which when received by a second node is acknowledged by the second node, wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for determining a quantity that is representative of the link utilization between the first and second nodes;

computer code for calculating a quantity that is representative of an average round transmission trip time for data to be sent from the first node to the second node and an acknowledgment to be received by the first node; and

computer code for calculating the average queue size function dependent on a data drop probability based upon the link utilization, the buffer size, and the average round trip transmission time; and

computer code for employing the values and parameters to define a drop probability which increases gradually in an overload condition.

75. (original) A computer program product according to claim 74, wherein the average queue size function is dependent upon the number of flows through the queue and an average packet size.

76. (original) A computer program product according to claim 75, further comprising:

predicting a drop probability based in part upon the average queue size function.

77. (original) A computer program product according to claim 76, wherein the average queue size function is dependent upon the number of flows through the queue and an average packet size.

78. (previously presented) A computer program product for estimating an average queue size for a node having a buffer with a queue wherein the node resides on a link, wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for determining a round trip transmission time for the link; and

computer code for determining the average queue size at the intersection point of a node congestion control function and a queue law function if there is full link utilization, wherein the queue law function is based in part on the round trip transmission time, and wherein the control function defines a drop probability which increases gradually in an overload condition.

79. (previously presented) A computer program product for designing a control function for use in a congestion control module residing in a network, wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for determining a maximum average queue size function based at least upon a minimum value for the average round trip transmission time;

computer code for selecting a point defining a maximum value for the control function outside of the maximum average queue size function; and

computer code for defining the control function as being bounded by the maximum value and crossing the maximum average queue size function, the control function defining a drop probability which increases gradually in an overload condition.

80. (original) A computer program product according to claim 79, wherein the computer code for defining a function includes selecting a linear equation as the control function wherein the linear function passes through the maximum value point.

81. (original) A computer program product according to claim 79, wherein computer code for selecting the point is also dependent on a queue management policy.

82. (original) A computer program product according to claim 79, wherein the maximum control function is dependent upon line speed for the network.

83. (original) A computer program product according to claim 79, wherein the control module is based on a random early detection control function.

84. (original) A computer program product according to claim 79, further comprising:

computer code predicting a drop probability based in part upon the average queue size function in a congestion control module of the first link.

85. (previously presented) A computer program product for determining parameters used in a random early detection congestion control module residing in a node in a network, wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for receiving input parameters including a line speed for the node; and

computer code for calculating values including a buffer size for an input to the link, a queue sampling interval, and an average weight; and

computer code for employing the values and parameters to define a drop probability which increases gradually in an overload condition.

86. (original) A computer program product according to claim 85, wherein the values for configuring the algorithm are used to determine values for qmin, pmin, qmax and pmax.

87. (previously presented) A computer program product for determining the minimum buffer size in a congestion control module having a control function in a TCP network defined by a queue law, wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for determining an equilibrium point where the control function and the queue law intersect, the control function defining a drop probability which increases gradually in an overload condition; and

computer code for selecting a buffer size that is larger than the average queue size at the intersection point.

88. (previously presented) A computer program product for creating a stable queue control function for managing a queue in a node within a network, wherein the queue control function determines a packet drop rate based upon an average queue size, wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for calculating a maximum queue law function based on traffic conditions; and

computer code for designating a maximum boundary for expected operating conditions of the queue control function to be outside of the maximum queue law function, wherein the control function defines a drop probability which increases gradually in an overload condition.

89. (original) A computer program product according to claim 88, wherein the queue control function is a random early detection control function.

90. (previously presented) A computer program product for setting a value for the maximum boundary point for expected operating conditions for a congestion control function in a network, wherein the computer program product has computer code on a computer readable medium, the computer code comprising:

computer code for selecting a queue management policy;

computer code for determining a maximum average queue size for expected operating conditions based upon the selected queue management policy;

computer code for selecting a corresponding value for the drop rate to be any point which lies outside of a queue law function for the network, thereby defining a drop probability which increases gradually in an overload condition.

91. (original) A computer program product according to claim 90, wherein the queue management policy is a drop conservative policy.

92. (original) A computer program product according to claim 90, wherein the queue management policy is a delay conservative policy.